

LISTING OF THE CLAIMS

Please amend Claims 1, 10 and 15 as indicated below.

1. (Currently Amended) An exception handling mechanism comprising:
an exception handler for recording exception information dependant on
types of exceptions and programming tasks that encounter
exceptions; and

a recovery agent for taking an action upon an occurrence of an exception
that occurred for a programming task, wherein the action is performed outside of
a debugging operation;

wherein the action to be taken upon the occurrence of the exception
corresponds to a type of exception and a programming task, and includes one or
a combination of restarting the programming task, terminating the programming
task, resetting a system running the programming task, and disregarding the
exception,

wherein the exception handler and the recovery agent run on a first
system operates autonomously and is embedded in a second system.

2. (Original) The mechanism of claim 1 wherein the recorded exception
information associated with an exception is associated with a signature for
identifying the recorded exception information with its associated exception.

3. (Original) The mechanism of claim 2 wherein the signature includes a
version of a program running the programming task.

4. (Original) The mechanism of claim 1 wherein a plurality of sets of
exception information for a plurality of exceptions is maintained in the system

running the programming task; each set of exception information being associated with a signature for identifying that set of exception information.

5. (Original) The mechanism of claim 1 wherein the recorded exception information associated with an exception is associated with a signature for identifying the format of the exception information.

6. (Previously Presented) The mechanism of claim 1 wherein the recorded exception information includes data related to a program stack, including data to reconstruct the program stack at time of exception.

7. (Original) The mechanism of claim 1 further comprising an analysis tool communicating via an interface with the system running the programming task, for identifying causes of the exception.

8. (Original) The mechanism of claim 7 wherein the analysis tool uses a version to match the object code of a program running the programming task to the source code of the program.

9. (Cancelled)

10. (Currently Amended) A processing system comprising:
a first system;

[[a]] an autonomous second system embedded in the first system;

an exception handler running in the second system for recording exception information upon an occurrence of an exception in the second system;
and

a recovery agent running on the second system, for taking an action upon the occurrence of the exception based on the recorded exception information, wherein the action is performed outside of a debugging operation;

wherein the action corresponds to a type of exception ~~[[and]]~~ that occurred in a programming task.

11. (Original) The processing system of claim 10 further comprising an analysis tool for receiving, via an interface, the recorded exception information from the second system and for identifying the cause of the exception.

12. (Original) The processing system of claim 10 wherein the second system includes nonvolatile memory for storing exception information.

13. (Original) The processing system of claim 12 wherein the exception information stored in the non-volatile memory is compressed.

14. (Original) The processing system of claim 12 wherein the exception information stored in non-volatile memory includes a plurality of sets of exception information, each set being associated with an exception and a signature.

15. (Currently Amended) A computing system comprising:
an exception handler for recording exception information on non-volatile memory upon an occurrence of an exception;

a recovery agent for taking an action upon the occurrence of the exception based on the recorded exception information, wherein ~~[[said]]~~ the action is performed outside of a debugging operation; and

an analysis tool for identifying the cause of the exception;
wherein the analysis tool receives the exception information from the nonvolatile memory via an interface interfacing a first system and a second system running the exception handler and the recovery agent wherein the second system is embedded in a third system and the second system operates autonomously of other systems.

16. (Cancelled).

17. (Original) The computing system of claim 15 wherein the recorded exception information includes data related to a program stack.